

Сървър – Клиент команди - SSI

- [Въведение](#)
- [Команди](#)
- [Примери](#)

SSI - Server Side Includes

Въведение

Технологията **SSI** позволява изпълнението в реално време с относително прости средства на примитиви (команди) върху HTTP сървъра. Резултатът от изпълнението е HTML страница, визуализирана при клиента. Технологията не е обект на стандартизиране, но се поддържа от по-разпространените HTTP сървъри. Възможностите на **SSI** и неговите настройки **SSI+** включват извеждане на дата, броя на посетителите, зареждане на страници по условие, изпращане на електронна поща, справки от бази данни и др.

Проблем със съвместимост на **SSI** с разпространените Web браузри не съществува, понеже командите се интерпретират и изпълняват от HTTP сървър (Server Site) вмъкнати (Includes) в HTML страница, а резултатът от интерпретацията на сървъра се формира като HTML код. Проблем може да възникне, ако самият HTTP сървър не поддържа тази технология или не е недостъпна за клиента поради липса на съответните привилегии. Някои сървъри може да поддържат само определено множество от **SSI** команди, което е отразено в тяхната документация.

За използване на **SSI** технология е необходимо разширението на HTML документа, съдържащ **SSI** команди, да бъде **.shtml**, вместо **.html**. Разширения **.sht** и **.shtm** също са приемливи, но това зависи от HTTP сървъра. Страница с това разширение предизвиква в HTTP сървър, съвместим с **SSI**, процес на претърсването ѝ за команди, които трябва да бъдат изпълнени преди изпращането на страницата при клиента за визуализация от неговия браузър. Този процес в сървъра е познат като *parsing*.

HTTP сървърите на *Netscape*, *Microsoft* и *Apache* поддържат технологията **SSI**, но множеството от разпознавани команди не е еднакво.

Команди

Общият формат на командите в **SSI** технологията е:

```
<!--#<tag><var> -->
```

Използваните елементи са: **break**, **config**, **echo**, **email**, **exec**, **if**, **include**, **fsize**, **flastmod**, **label**, **goto** и **odbc**. Резервирана "дума" е комбинацията "&&", поставена пред име на променлива, предизвиква нейното преизчисление последвано от изпълнение на съответната **SSI** команда.

Всяка команда започва с коментарен низ "<!--", което е и своеобразна форма на защите при наличие на несъвместимости, т.е. тези редове няма да се интерпретират и да предизвикат страничен ефект при сървъра или брауъра. Индикация към сървъра, че следва **SSI** команда е символът "#", поставен веднага след коментара. Така, всяка **SSI** команда започва с "<!--#" и завършва с " -->" (*внимание*, предхожда се от шпация). Някои сървъри възприемат завършващия елемент и без шпация, но правилото е да има интервал между съдържанието на командата и завършващият я елемент.

break

Елементът играе ролята на терминиращ знак (край) за една HTML страница. Елементът може да се използва в комбинация с други за изпращане на части от страница при клиента в зависимост от определени условия.

Синтаксис:

```
<!--#break -->
```

config

Елементът позволява конфигуриране или създаване на шаблони при извеждане на информация от други **SSI** елементи. За целта се използват параметри:

- **Errmsg** - задава изходен файл със съобщения за грешки от анализа на **SSI** команди.
- **Timefmt** - задава изходен формат за изобразяване на време.
- **Sizefmt** - задава формат за изобразяване размера на файл.
- **Cmdecho** - включва/изключва стандартния изход при изпълнение на команда **exec**.
- **Cmdprefix** - дефинира символен низ като префикс за всеки ред от изпълнението на команда с **exec**.
- **Cmdpostfix** - дефинира символен низ като постфикс за всеки ред от изпълнението на команда с **exec**.
- **Onerr** - дефинира действието при възникване на условие по изключение (грешка):
 - **goto label** - преминава към етикет;

- ***print texte*** - извежда текст, може и HTML;
- ***error*** - извежда съобщение за грешка;
- ***break*** - прекъсва HTML страница;
- ***errorbreak*** - прекъсва HTML страница със съобщение за грешка;
- ***printbreak texte*** - прекъсва HTML страница и извежда текст, може и HTML.

Шаблони за изобразяване на дата и час от елемента *echo*.

Разширен формат	Съкратен формат
%A - Пълно изписване на деня от седмицата	%a - Съкратено изписване на деня от седмицата
%B - Пълно наименование на месеца	%b - Съкратено наименование на месеца
%Y - Изписва годината в четири знака	%y - Изписва годината в два знака
%D - Дата с формат mm/dd/yy	%d - Ден от месеца
%H - Извежда часа от 1 до 24	%i - Извежда часа от 1 до 12
%M - Извежда минутите от 0 до 60	%m - Извежда месеците от 01 до 12
%R - Извежда часа като %H: %M	%r - Извежда часа като %I: %M: %S: %p
%p - Добавя към часа a.m. или p.m.	%T - Извежда часа като %H: %M: %S
%S - Извежда секундите от 0 до 60	%Z - Времева зона

echo

Елементът *echo* позволява вмъкването на данни в една HTML. Асоциираната към елемента променлива *var* трансформира стойността си в символен низ, който я замества в HTML кода на страницата.

Синтаксис:

```
<!--#echo var=cName -->
```

Променливата *var* може да бъде определена като изход от формуляр чрез метода *POST* или чрез глобална променлива. При първия случай на променливата *var* се присвоява името на съответната променлива от формуляра. Във втория случай,

списъкът от глобални променливи зависи от HTTP сървъра. Той може да бъде по-богат или по-кратък. Често използваните променливи са следните:

- **ACCEPT_LANGUAGE** - списък от езици разпознавани от клиента.
- **DATE_GMT** - текущата дата и час на сървъра според GMT (*Greenwich Meridian Time*).
- **DATE_LOCAL** - локалната дата и час на HTTP сървъра.
- **DOCUMENT_NAME** - пълното име на документа.
- **DOCUMENT_URI** - относителния адрес на документа според корена на сървъра.
- **FROM** - електронен адрес на клиента.
- **FORWARDED** - име на проху server, който е пренасочил клиента.
- **GATEWAY_INTERFACE** - име и версия на CGI модула на HTTP сървъра.
- **HTTP_ACCEPT** - разпознаван от клиента тип MIME.
- **HTTP_COOKIE** - cookies в HTTP сървъра.
- **HTTP_USER_AGENT** - името на Web браузъра при клиента.
- **LAST_MODIFIED** - дата и час на последната модификация на HTML документа.
- **REMOTE_ADDR** - IP адреса на клиента.
- **REMOTE_IDENT** - името на отдалечения потребител, според нормата RFC931.
- **REMOTE_HOST** - име на клиентската машина.
- **REQUEST_METHOD** - метод за искане на документа.
- **SERVER_NAME** - име на машината с инсталиран HTTP сървър.
- **SERVER_PORT** - номер на port за HTTP сървъра.
- **SERVER_PROTOCOL** - име и версия на HTTP сървъра.
- **SERVER_SOFTWARE** - наименование на HTTP сървъра.

email

Елементът *email* позволява изпращане на електронна поща при достъп до определена страница или от формуляр. Параметрите на елемента са следните:

- **fromhost** - задава името на компютъра подател на съобщението (*задължителен*).
- **tohost** - задава името на SMTP сървъра (*задължителен*).
- **fromaddress** - задава електронния адрес на подателя (*задължителен*).
- **toaddress** - определя електронния адрес на получателя (*задължителен*).
- **message** - определя съдържанието на съобщението.
- **message** - дефинира предмета на съобщението.
- **sender** - дефинира адреса на изпращача.
- **replyto** - определя обратния адрес за съобщението.
- **cc** - задава полето CC:
- **inreplyto** - определя полето inreplyto.
- **id** - определя номер, идентифициращ съобщението.

exec

Елементът *exec* позволява изпълнението на произволна команда от операционната система на сървъра.

Синтаксис:

```
<!--#exec cmd|cgi -->
```

Променливите могат да бъдат:

- **cmd** - задава името и параметрите за изпълнимата програма на HTTP сървъра.
- **cgi** - задава името на CGI модула.

filesize

Елементът *filesize* позволява вмъкването на размер на файл като параметър, заместващ позицията на **SSI** командата в HTML страница.

Синтаксис:

```
<!--#filesize virtual|File="file_name" -->
```

Променливите могат да бъдат:

- **virtual** - променливата позволява указване на относителен път спрямо корена на HTTP сървъра.
- **file** - променливата позволява указване на относителен път спрямо текущия HTML документ.

lastmod

Елементът *lastmod* позволява вмъкването на датата на последна модификация на файл като параметър, заместващ позицията на **SSI** командата в HTML страница.

Синтаксис:

```
<!--#lastmod virtual|File="file_name" -->
```

Променливите могат да бъдат:

- **virtual** - променливата позволява указване на относителен път спрямо корена на HTTP сървъра.

- **file** - променливата позволява указване на относителен път спрямо текущия HTML документ.

goto

Елементът *goto* позволява преход към етикет (*#label*) без да бъдат изпълнявани програмните редове между елемента и етикета.

Синтаксис:

```
<!--#goto = "'label'" -->
```

if

Елементът *if* позволява изпълнението на определено действие при определени условия.

Синтаксис:

```
<!--#if "операнд1" оператор "операнд2" операция -->
```

където:

- **операнд1** - първи операнд в сравнението.
- **операнд2** - втори операнд в сравнението.
- **оператор** - сравнение, изразено чрез символите : '==', '!=', '<', '>', '>' или '<'.
- **операция** - е действие, което се изпълнява, ако при теста на условието се докаже неговата истинност : **goto**, **print**, **error** или др.

include

Елементът *include* позволява вмъкване на съдържанието на файл в HTML страница. **SSI+** технологията е рекурсивна. Всеки вмъкнат документ може да съдържа **SSI+** елементи, които да предизвикват вмъкването на други файлове.

Синтаксис:

```
<!--#include virtual|File="file_name" -->
```

Променливите могат да бъдат:

- **virtual** - променливата позволява указване на относителен път спрямо корена на HTTP сървъра.

- **file** - променливата позволява указване на относителен път спрямо текущия HTML документ.

label

Елементът *label* задава позицията на етикет, използван за преход от елемент *goto*.

Синтаксис:

```
<!--#label = "label_name" -->
```

odbc

Елементът *odbc* позволява изпращане на заявки към база данни. За целта са налице няколко *odbc* елемента:

- **connect** - осъществява връзка със съществуваща база данни.

Синтаксис:

```
<!--#connect="datasource,username,password" -->
```

където:

datasource - е odbc името на БД, регистрирано с ODBC Utility;

username - е името на регистриран потребител;

password - е паролата за достъп на този потребител.

- **debug** - позволява диагностика на механизма ODBC и SSI. Съобщенията се появяват в тялото на HTML страницата.

Синтаксис:

```
<!--#debug низ -->
```

- **format** - задава шаблона за изобразяване на получените от БД данни от заявката.

Синтаксис:

```
<!--#format="списък от шаблони" -->
```

- **statement** - задава SQL заявка към БД.

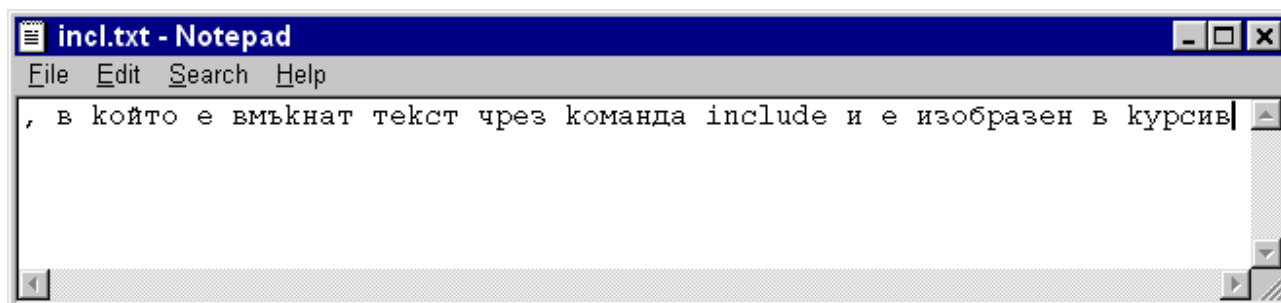
Синтаксис:

```
<!--#statement="SQL заявка" -->
```

Пример:

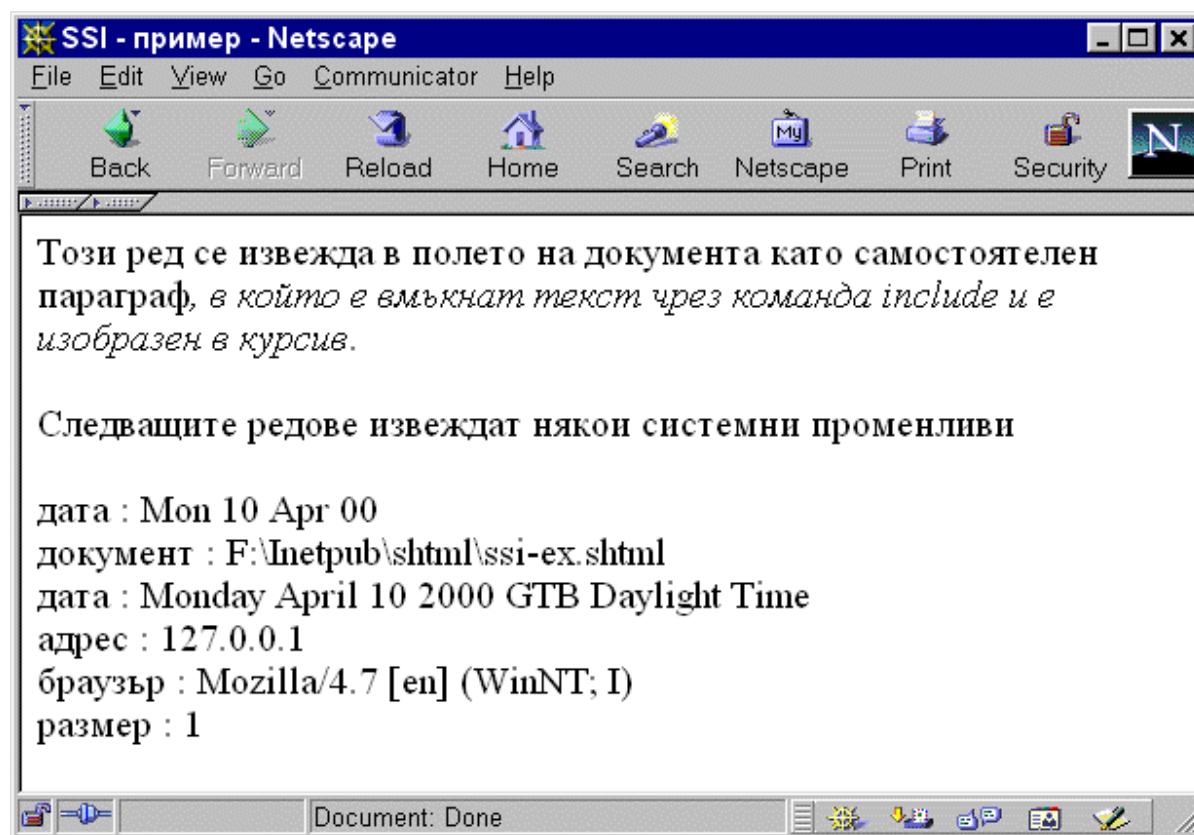
```
<html>
<head><title>SSI - пример</title></head>
<body>
<p>Този ред се извежда в полето на документа като самостоятелен параграф<i><!--
#include file="incl.txt" --></i>.
<p>Следващите редове извеждат някои системни променливи</p>
Дата : <!--#config timefmt="%a %d %b %y" --><!--#echo var="DATE_GMT"--><br>
Документ : <!--#echo var="DOCUMENT_NAME"--><br>
Дата : <!--#config timefmt="%A %B %d %Y %R %Z" --><!--#echo var="DATE_LOCAL"--
><br>
Адрес : <!--#echo var="REMOTE_ADDR" --><br>
Браузър : <!--#echo var="HTTP_USER_AGENT" --><br>
Размер : <!--#fsize virtual="shtml\break.shtml" --><br>
</body>
</html>
```

Кодът съдържа команди общи за почти всички HTTP сървъри поддържащи **SSI** технология, В примера с команда *include* се вмъква в документа съдържанието на файл *incl.txt*, чието съдържание е показано на Екран 4.



Екран 4 - Съдържание на файл *incl.txt*

Резултатът от интерпретацията на примера е показан на Екран 5.



Екран 5 - Пример за SSI технология

Примерът е достъпен за изпълнение в режим on-line при избор на тази връзка.

Интернет за персонални компютри
Copyright © 1998-2001 by Ivan Madjarov All rights reserved